(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 651 328 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **94116832.0**

(51) Int. Cl.[5] **G06F 9/46**

(22) Date of filing: **25.10.94**

(30) Priority: **27.10.93 US 143870**

(43) Date of publication of application:
03.05.95 Bulletin 95/18

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **MICROSOFT CORPORATION**
One Microsoft Way
Redmond,
Washington 98052-6399 (US)

(72) Inventor: **Seaman, Michael R. C.**
237-6th Avenue
Kirkland,
Washington 98033 (US)
Inventor: **Ross, Kevin W.**
27826 N. E. 144th
Duvall,
Washington 98052 (US)
Inventor: **Blanford, Mark S.**
11008-178th Court N.E.
Redmond,
Washington 98052 (US)
Inventor: **Heizer, Isaac J.**
16904 N.E. 176th Street
Woodinville,
Washington 98072 (US)

(74) Representative: **Patentanwälte Grünecker, Kinkeldey, Stockmair & Partner**
Maximilianstrasse 58
D-80538 München (DE)

(54) Event architecture for system management in an operating system.

(57) An event system is provided within an object-oriented environment. The event system informs users and system functions of events within the system. Events may be modeled as objects that are visible within the global namespace. These objects include event source objects and event sink objects. Event source objects generate event reports and event sink objects are the objects that receive reports. Special objects may be incorporated in the system to direct event reports from an event source object to an event sink object.

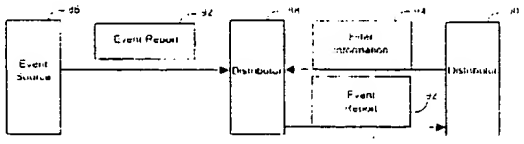

Figure 8

EP 0 651 328 A1

report from the event source object. As part of the registration, an instance of a function in an interface to be called by the event report as received by the event sink object is specified.

Brief Description of the Drawings

Figure 1 is a block diagram of a data processing system that is suitable for practicing a preferred embodiment of the present invention.

Figure 2A is a block diagram illustrating an instance of the preferred embodiment in which a single event report is sent from a single event source to a single event sink.

Figure 2B is a block diagram illustrating an instance of the preferred embodiment in which a single event report is sent from a single event source to multiple event sinks.

Figure 2C is a block diagram illustrating an instance of the preferred embodiment in which multiple event sources send multiple event reports to a single event sink.

Figure 2D is a block diagram illustrating an instance of the preferred embodiment in which a single event source sends multiple event reports to multiple event sinks.

Figure 3 is a diagram of the run time data structures used for interfaces by the preferred embodiment to the present invention.

Figure 4 is a diagram of objects that play a role in the preferred embodiment of the present invention to register an event sink object with an event source object to receive event reports.

Figure 5 is a flowchart illustrating the steps performed by the preferred embodiment to register an event sink object with an event source object to receive event reports.

Figure 6 is a flowchart illustrating steps performed by the preferred embodiment of the present invention in generating and processing event reports.

Figure 7 is a block diagram of a preferred embodiment of the present invention when a distributor object is employed.

Figure 8 is a block diagram illustrating propagation of an event rePort from a first distributor object to a second distributor object in accordance with a preferred embodiment of the present invention.

Detailed Description of the Invention

A preferred embodiment of the present invention provides an event system architecture for generating system management events. The architecture is designed to be easily used and to require minimal overhead. The system management events notify processes of certain states.

Figure 1 shows an illustrative data processing system 10 for practicing a preferred embodiment of the present invention. Those skilled in the art will appreciate that other types of data processing systems may be used for practicing the present invention. The data processing system 10 includes a central processing unit (CPU) 12 that oversees activities of the system. The CPU 12 communicates with a memory 14, a keyboard 16, a video display 18, a printer 19 and a mouse 24. The memory 14 holds an object-oriented operating system 20 that includes an event system 22. Although the operating system 20 of the preferred embodiment is an object-oriented operating system, those skilled in the art will appreciate that the present invention is not limited to such an object-oriented environment. The keyboard 16 and mouse 24 are conventional input devices, and the video display 18 and printer 19 are conventional output devices.

The event system 22 is responsible for overseeing the generation and transmission of event reports that report the occurrence of particular events in the system 10. An "event," in this context, is an asynchronously arising condition relative to a destination process that wishes to be informed of the event. The process in which the condition arises is the "event source," whereas the process to which the event is reported is the "event sink." An "event report" reports an event and is transmitted from the event source to the event sink (as will be described in more detail below). The event source and event sink are objects that are visible in a distributed namespace of the system 10. The distributed namespace is a logical organization of the "names of objects" (described in more detail below) that are stored in the system 10.

The preferred embodiment of the present invention allows application programs run on the system 10 (Figure 1) to generate and receive information about external events without possessing knowledge about the events or the sources of the events. It provides a means for an event sink 27 (Figure 2A) to register with an event source 23 to receive an event report 25. The event sink 27 may then invoke an event handler to respond to the occurrence of the event. The design of the preferred embodiment leverages the distributed namespace of the system to provide places for event sinks to register. The preferred embodiment also provides a means to send the event report from the event source to the event sink.

preferred embodiment of the present invention to establish connections between an event source and an event sink. This process will be described below as "registration".

Before discussing how an event sink registers with an event source to receive event reports, it is helpful to consider how events are defined relative to the objects that may generate them. In general, an object specifies an event set of events that it is capable of generating. The event set is part of the object's definition and is described in type information provided for the object. The type information may be stored in a storage structure that is separate from the object. The type information specifies which events are provided in the objects of that set. Consider as an example as object that supports an event set known as the DiskEventSet. This event set is identified as follows:

```
{
        DiskSpaceLow( DiskSpaceEvent psEvent);
        DiskFull( DiskSpaceEvent psEvent);
        DiskError( DiskErrorEvent psEvent)
}
```

DiskSpaceEvent is a property set that may be defined as follows:

```
propset DiskSpaceEvent : public DiskEvent
    {
        LARGE_INTEGER       DiskSpaceAvailable;
        LARGE_INTEGER       TotalDiskSpace;
    }
propset DiskEvent : public ISystemEvent
    {
        ULONG               ulVolumeID;
        WCHAR *             pwzVolumeName;
    }
propset ISystemEvent
    {
    mandatory:
        SYSTIME             timeEventCreationTime;
        UUID                uuidClassId;
        ULONG               sevSeverity;
        UUID                uuidCategoryID;
        ULONG               ulEventCode;
        ULONG               ulHopCount;
    }
```

Before discussing the registration process in more detail, it is useful to first introduce the objects which play a role within this registration process. Figure 4 is a diagram illustrating an example situation wherein the objects that play a role in registration are used. Figure 4 illustrates an instance wherein a printer 46

A system may also include special objects, known as "distributor objects." A distributor object is an object which exists in the global namespace to route event reports to event sinks. Figure 7 depicts a distributor object 82 that receives event reports from event sources 80, 80' and 80". The distributor object 82 provides both event source and event sink functionalities. The distributor object 82 routes the event reports received from the event sources 80, 80' and 80" to event sinks 84, 84' and 84". Distributor objects 82 are useful in grouping event sources and sinks as sets. For example, a distributor may be used to represent all administrators on a given domain.

A distributor object may also be configured to propagate event reports to other distributor objects. Figure 8 shows an example of a situation in which a distributor object 88 receives an event report 92 from an event source 86 and propagates the event report to a second distributor object 90. The difference between this type of propagation and the propagation that occurs with normal registration is that filtering information 94 stored on the second distributor object 90 is forwarded to the first distributor object 88. This filtering information specifies which type of event report the distributor object 90 is interested in obtaining. The first distributor object 88 filters the incoming event reports 92 to determine whether the second distributor object 90 wishes to receive the event report. The filtering information 94 is stored on the second distributor object in such a way that it is available to any distributor object that is registered to propagate event reports to the second distributor object.

The operating system 20 provides a number of dispatch interfaces. A dispatch interface, in this context, is as defined in the Microsoft Object Linking and Embedding (OLE) 2.0 protocol, established by Microsoft Corporation of Redmond, Washington. Dispatch interfaces allow access to methods, properties and data members of objects whose interface is not known. Each dispatch interface has functions that allow a caller to retrieve and send property values. The event distributor objects have the ability to call any dispatchable interface. Parameter information is stored with the registration information for the event sink object. Properties are set on the software connector to the event sink in the event source such that each registration can have its own interface to be called. As an example, this capability allows a system event to trigger the printing of a word processing document through the call to the unique interface associated with the registration of an event sink.

It should be appreciated that registrations may be maintained over a period of time. Registrations are not strictly a one-time phenomena. The registration need not be deleted after a single event occurs. Maintaining the registration is helpful in performing functions such as logging. Logging involves recording event reports in a file to create a log. The log may be later viewed to provide a historical record of activity of a part of the system.

A preferred embodiment of the present invention enhances the ability of system administration functions to become aware of relevant events within the system. This capability stems, in large part, from segregating the occurrence of an event from the response to the event. The preferred embodiment also facilitates segregation by event type so that the system administrator may be aware of the different types of events and respond accordingly.

Since event reports are objects in the global namespace, both users and system components have access to the event reports. The event reports include important state information that may be used by users and system components alike. The user has the ability to discover what events a program is capable of generating and can register to receive notice of any such events.

While the present invention has been described with reference to a preferred embodiment thereof, those skilled in the art will appreciate that various changes in form and scope may be made without departing from the present invention as defined in the appended claims. For instance, the present invention need not be practiced in a single processor system like that shown in Figure 1; rather the present invention may also be practiced in a distributed system.

Claims

1. In a data processing system having memory means and processing means, a method comprising the steps of:

   storing a global namespace of objects in the memory means;

   providing event source objects for generating event reports in response to events at the event source objects, an event sink object for receiving event reports and a distributor object for distributing event reports from the event source objects to the event sink object;

   triggering an event at one of the event source objects;

   generating an event report at the event source object where the event was triggered;

   forwarding the event report to the distributor object; and

14. In a data processing system having memory means and processing means, a method comprising the steps of:

providing at least one event source object in the memory means for generating an event report in response to an event at the event source object;

providing a first distributor object and a second distributor object in the memory means, said first distributor object serving to distribute the event report when generated by the event source object to the second distributor object;

triggering the event at the event source object;

generating an event report at the event source object;

forwarding the event report to the first distributor object;

providing filtering information from the second distributor object to the first distributor object, said filtering information specifying a type of event report that the second distributor object wishes to receive; and

where the filtering information indicates that the second distributor object wishes to receive the event report, forwarding the event report from the first distributor object to the second distributor object.

15. In a data processing system having processing means and memory means, a method comprising the steps of:

providing an event source object for generating an event report in response to an event and an event sink object for receiving the event report; and

registering the event sink object to receive the event report from the event source object, wherein as part of the registration specifying an instance of a function in an interface to be called when the event report is received by the event sink object.

16. The method of claim 15, further comprising the steps of:

triggering the event at the event source object;

generating the event report at the event source object and forwarding the event report to the event sink object; and

in response to receiving the event report at the event sink object, invoking the function of the instance of the interface specified in the registration.

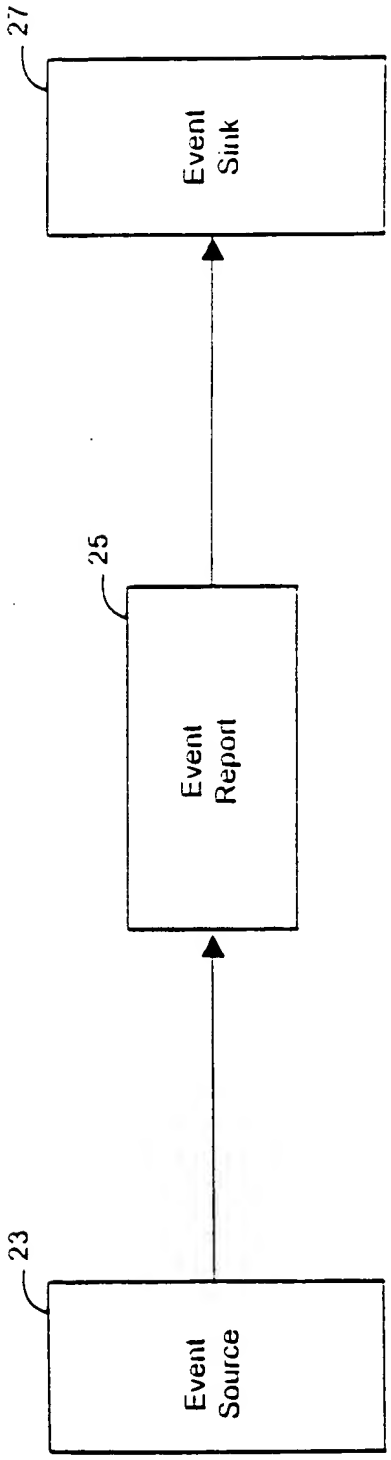

Figure 2A

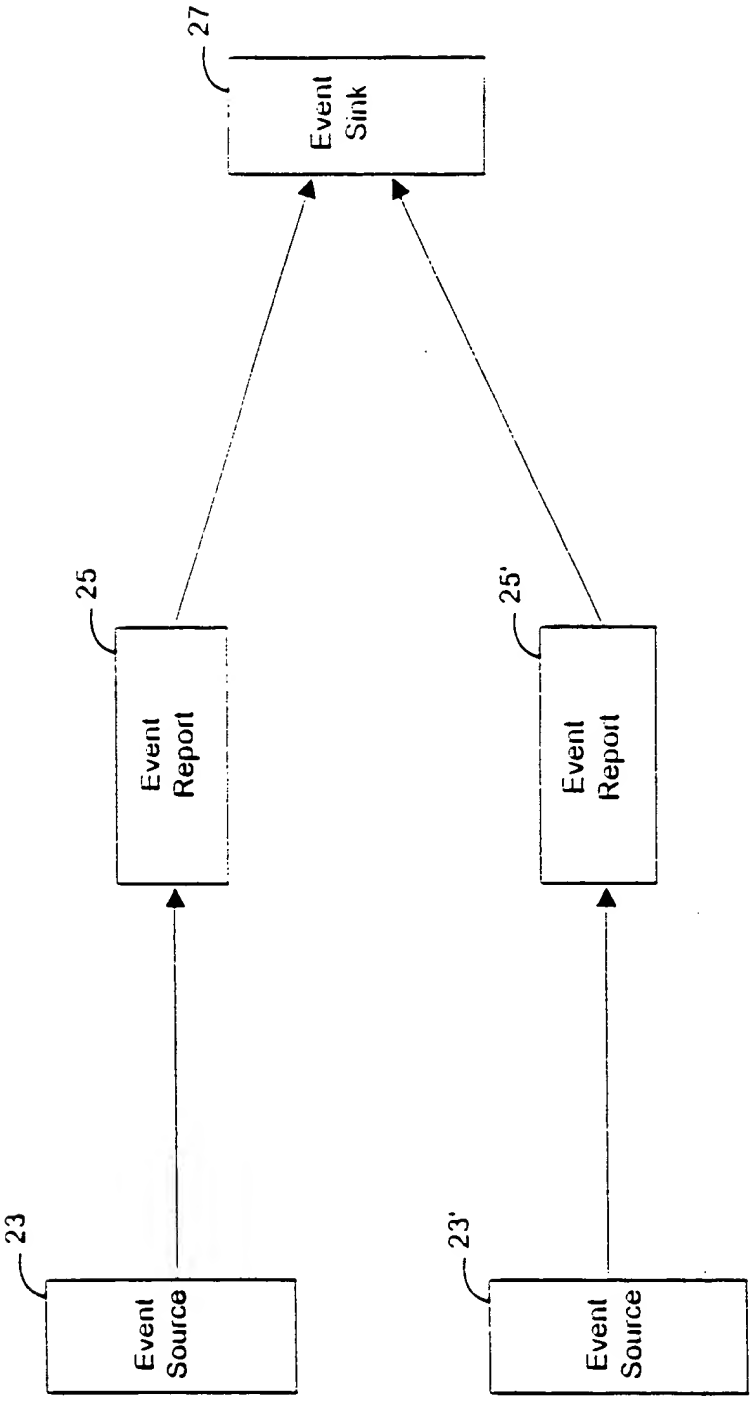
27
Event Sink
25
Event Report
25'
Event Report
23
Event Source
23'
Event Source

Figure 2C

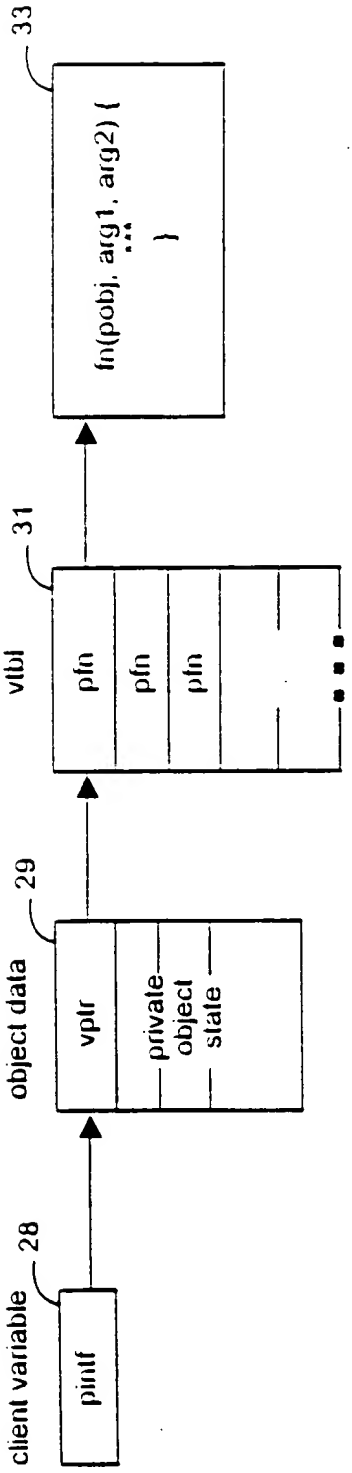

Figure 3

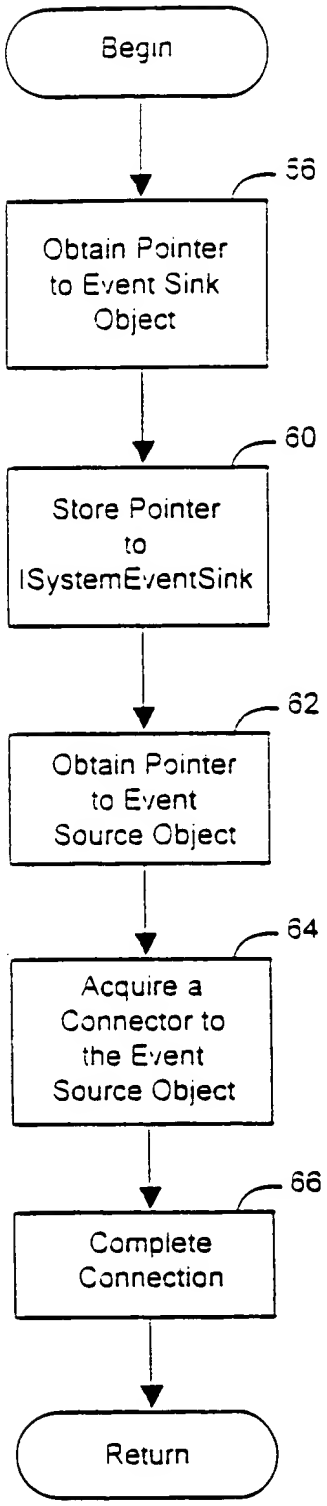
Begin
56
Obtain Pointer to Event Sink Object
60
Store Pointer to ISystemEventSink
62
Obtain Pointer to Event Source Object
64
Acquire a Connector to the Event Source Object
66
Complete Connection
Return

Figure 5

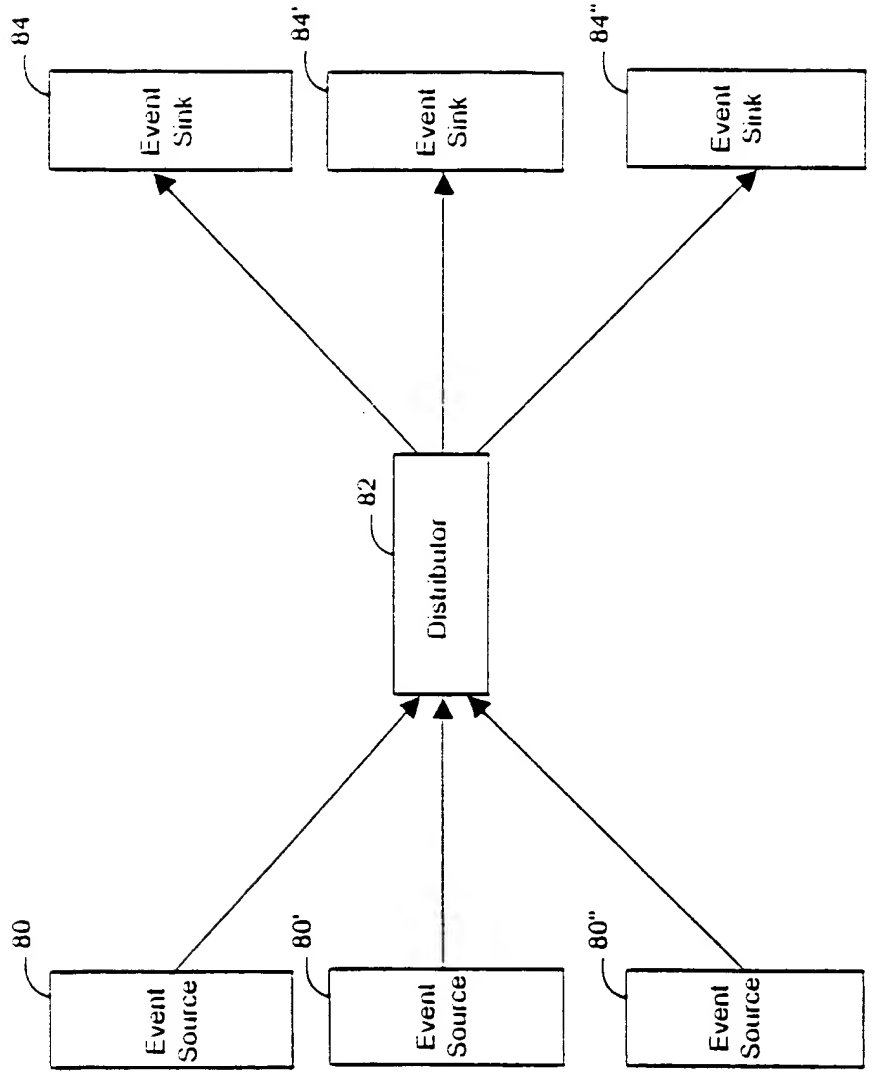

Figure 7

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 11 6832

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	WO-A-91 03017 (MICROSOFT CORPORATION) * page 2, line 35 - page 3, line 29; figure 1 *	1-16	G06F9/46
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol.34, no.4A, September 1991, NEW YORK US page 193 'Event handlers for an object-oriented OfficeVision'	5,14	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
			G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	9 February 1995	Corremans, G

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)